

NMOS Control & Monitoring

An Open Solution

Cristian Recoseanu – Tech Lead @ Pebble

Tech Lead for IS-12, BCP-008



[What NMOS is](#)

What:

A family of open, free of charge specifications that enable interoperability between media devices on an IP infrastructure.

Why:

Enables end-users and SIs to create best of breed solutions from a greater pool of vendors which interoperate at different layers

NMOS “layers”

Roadmap

Resource management

- Discovery and Registration (IS-04)
- Annotation (IS-13)
- Natural grouping (BCP-002-01)
- Asset Distinguishing Information (BCP-002-02)

Specs

Connection management

- Connection management (IS-05)
- Channel mapping (IS-08)
- Receiver capabilities (BCP-004-01)
- JPEG-XS (BCP-006-01)
- H264 (BCP-006-02)
- H265 (BCP-006-03)
- MPEG-TS (BCP-006-04)
- NDI (BCP-007-01)

Device Control & monitoring

- Event & tally (IS-07)
- Control protocol (IS-12)
- Control architecture (MS-05-01)
- Control framework (MS-05-02)
- Receiver status (BCP-008-01)
- Sender status (BCP-008-02)

Device Configuration

- Stream compatibility management (IS-11)
- Device configuration (IS-14)

Security

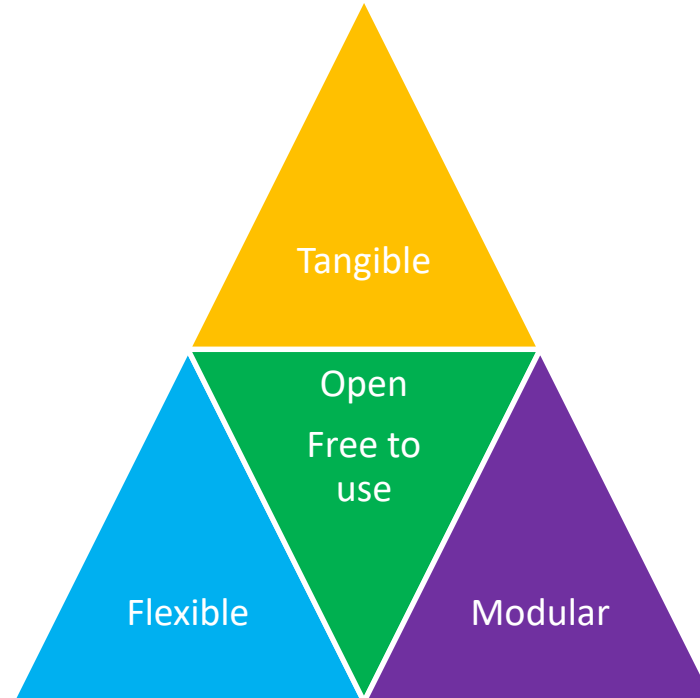
- Authorization API (IS-10)
- Secure comms (BCP-003-01)
- Authorization (BCP-003-02)
- Certificate provisioning (BCP-003-03)

NMOS C&M - An Open Solution

Establishes a standard, interoperable vision, philosophy and platform for device control and monitoring within the NMOS ecosystem and community.

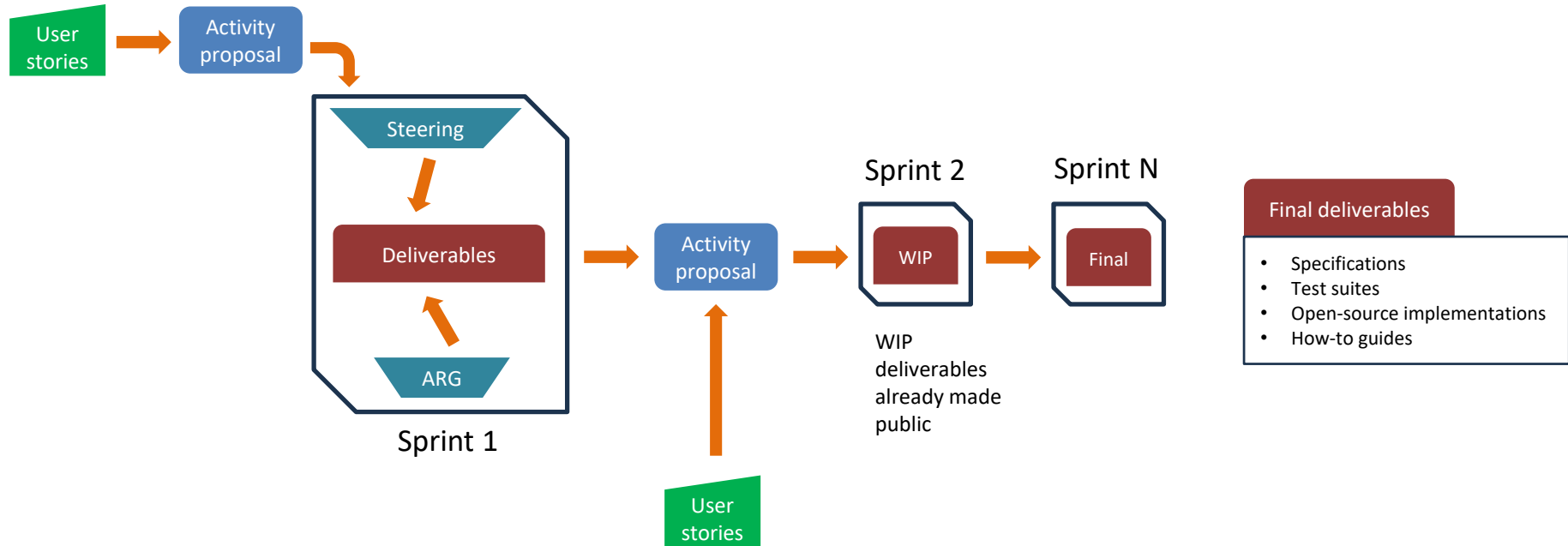
- Secure by design with [BCP-003](#) and [IS-10](#) specifying the requirements
- Architecture and roadmap are governed not by a single entity but by the NMOS community
- Benefits from [interoperability testing](#) within the NMOS ecosystem
- Benefits from a forum where vendors, end users and integrators can provide feedback about any concerns/improvements/integration issues they may have

Distinguishing attributes



Open

A truly open solution every step of the way.

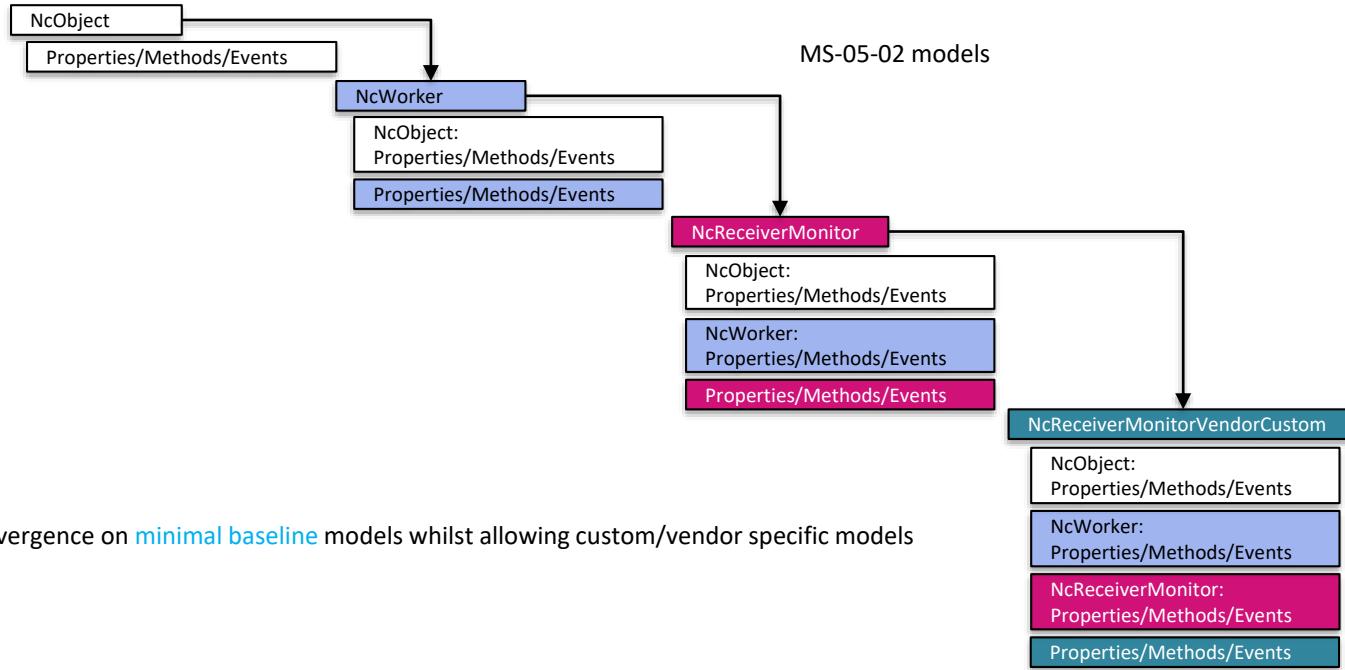


Flexible

MS-05-01	Architecture <ul style="list-style-type: none">• Vision• Philosophy• Overview
MS-05-02	Framework <ul style="list-style-type: none">• Modelling language & rules• Core control classes & datatypes portfolio• Device control model discovery
IS-12	Protocol <ul style="list-style-type: none">• Exposes and interacts with objects and properties• Commands and notifications• Transport and message encoding
BCPs	Feature sets Opt-in models and requirements for specific features
BCP-008-01	Receiver status Describes the status monitoring domains along with expectations, behaviour and conformance requirements
BCP-008-02	Sender status Describes the status monitoring domains along with expectations, behaviour and conformance requirements

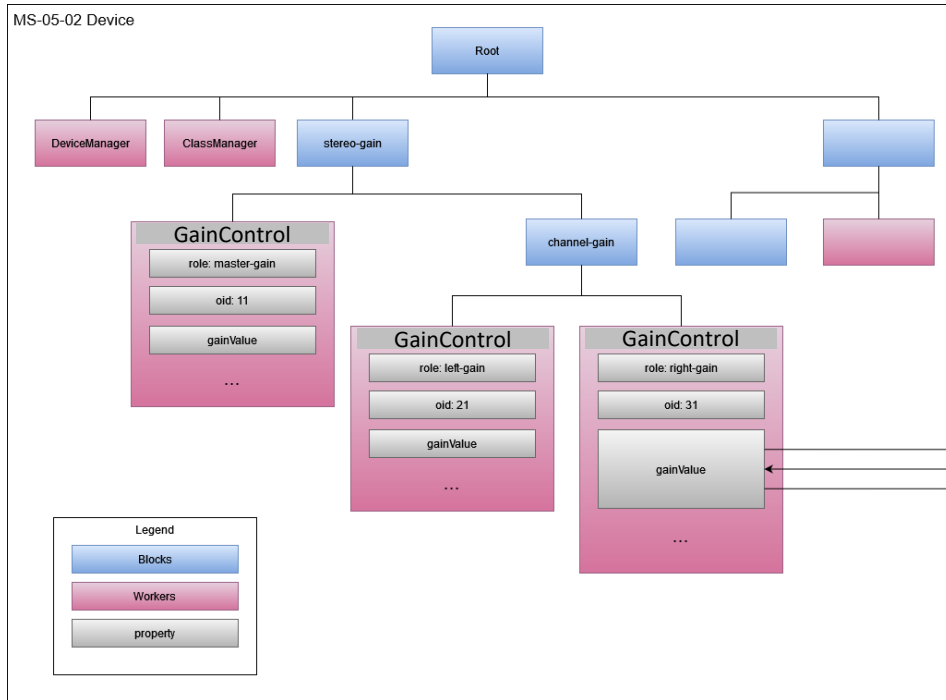
The problem space is explored at different levels offering the optimal amount of standardization whilst maintaining vendor freedom and ensuring interoperability.

Flexible



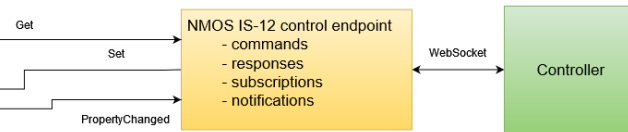
Convergence on [minimal baseline](#) models whilst allowing custom/vendor specific models

Flexible



```
{
  "messageType": 0,
  "commands": [
    {
      "handle": 2,
      "oid": 7777,
      "methodId": { "level": 1, "index": 2 },
      "arguments": { "id": { "level": 1, "index": 25 }, "value": 5 }
    }
  ]
}
```

IS-12 messages are compact but still readable



Convergence on the protocol

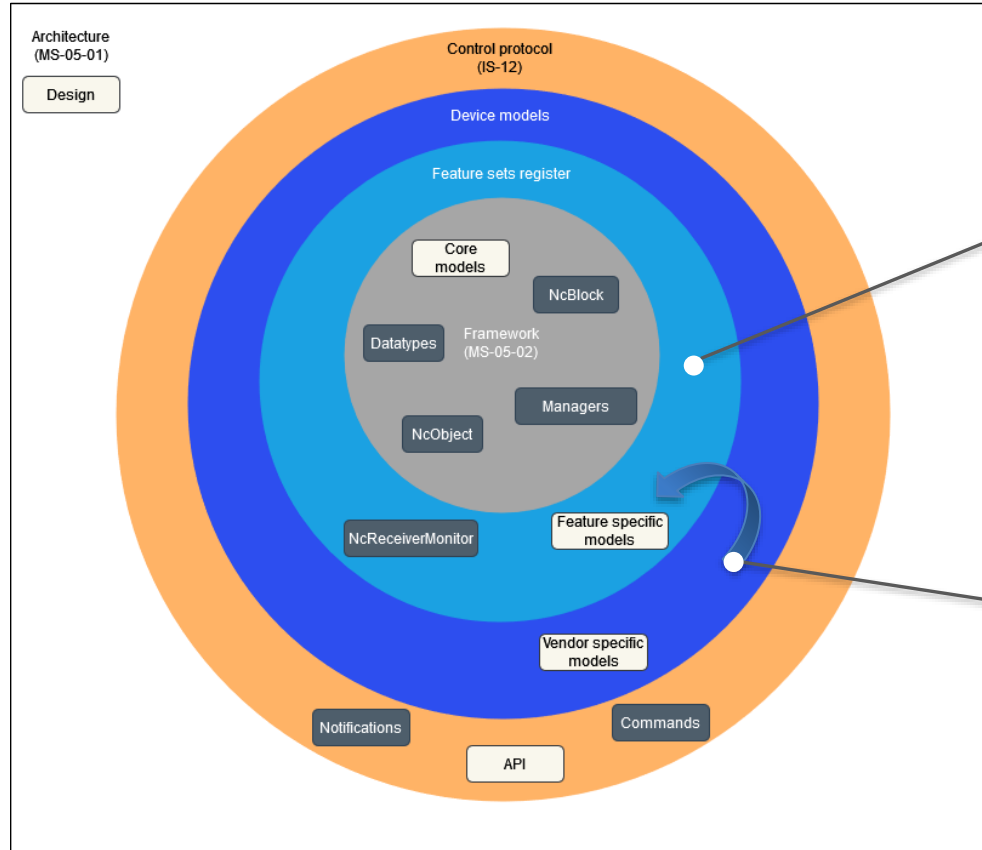
Flexible

The screenshot displays the 'control' interface with a sidebar on the left containing navigation links: Alerts (1), Home, Host info, Manage, and Panels. The main area is titled 'Manage / Control / Device explorer' and includes a 'Show help' link. Below the title, there are filters for 'Device: NC-01 device', 'Configuration: NMOS IS-12', and 'Profile: NMOS', along with a '+ New profile' button. The 'Device Tree' tab is active, showing a hierarchical view of the device structure. The tree starts with 'root: root', which contains 'DeviceManager: Device manager', 'ClassManager: Class manager', and 'receivers: Receivers'. Under 'receivers', there is a 'monitor-01: Receiver monitor 01' entry, which is expanded to show several properties: 'connectionStatus', 'connectionStatusMessage', 'payloadStatus', 'payloadStatusMessage', 'enabled', and 'classId'. The 'Details' panel on the right shows the selected property 'connectionStatus' with its path 'root/receivers/monitor-01'. It lists various metadata such as Label, Description, Factory Label, Factory Description, Local id, Value type, Is readonly, Is nullable, Is array, Data type name, RAW Value, and Possible values. The 'Possible values' section is a table with 4 rows.

Controller fully discovering an
MS-05/IS-12 device
(including vendor specific elements)

Value	Name
0	Undefined
1	Connected
2	Disconnected
3	ConnectionError

Flexible



More convergence around common feature models with **direct vendor and end user involvement**

Vendors propose existing vendor specific models for registration as a common feature set

Tangible

- All the deliverables end up in the public domain on GitHub
- WIP versions of the specifications are available publicly on GitHub from the very first few sprints

Tangible

- MS-05-01: NMOS Control Architecture
<https://specs.amwa.tv/ms-05-01/>
- MS-05-02: NMOS Control Framework
<https://specs.amwa.tv/ms-05-02/>
- IS-12: NMOS Control Protocol
<https://specs.amwa.tv/is-12/>
- BCP-008-01: Receiver status
<https://specs.amwa.tv/bcp-008-01/>
- BCP-008-02: Sender status
<https://specs.amwa.tv/bcp-008-02/>

Tangible

Developer resources and tools:

- INFO-006: Implementation guide for NMOS Device Control
<https://specs.amwa.tv/info-006/>
- NMOS Device control mock
<https://github.com/AMWA-TV/nmos-device-control-mock>
- nmos-cpp: Open-source node implementation
<https://github.com/sony/nmos-cpp>

Tangible

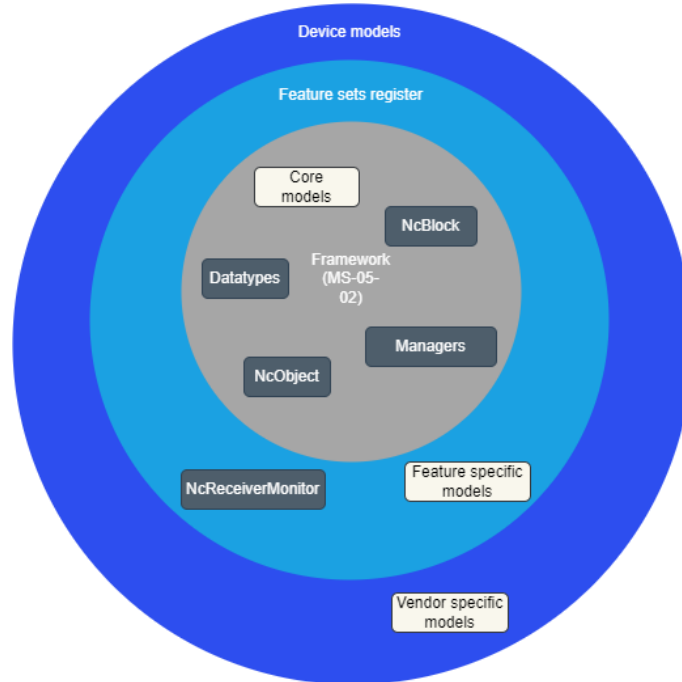
Comprehensive [test suite](#) covering:

- MS-05-02 framework core model conformance
- IS-12 conformance including commands, responses, subscriptions, notifications and error reporting
- Feature sets model conformance testing where we can opt in each individual feature set through configuration
- Behaviour testing for specific features defined in a BCP (BCP-008-01/02)
- Vendor specific models to ensure compatibility and interoperability

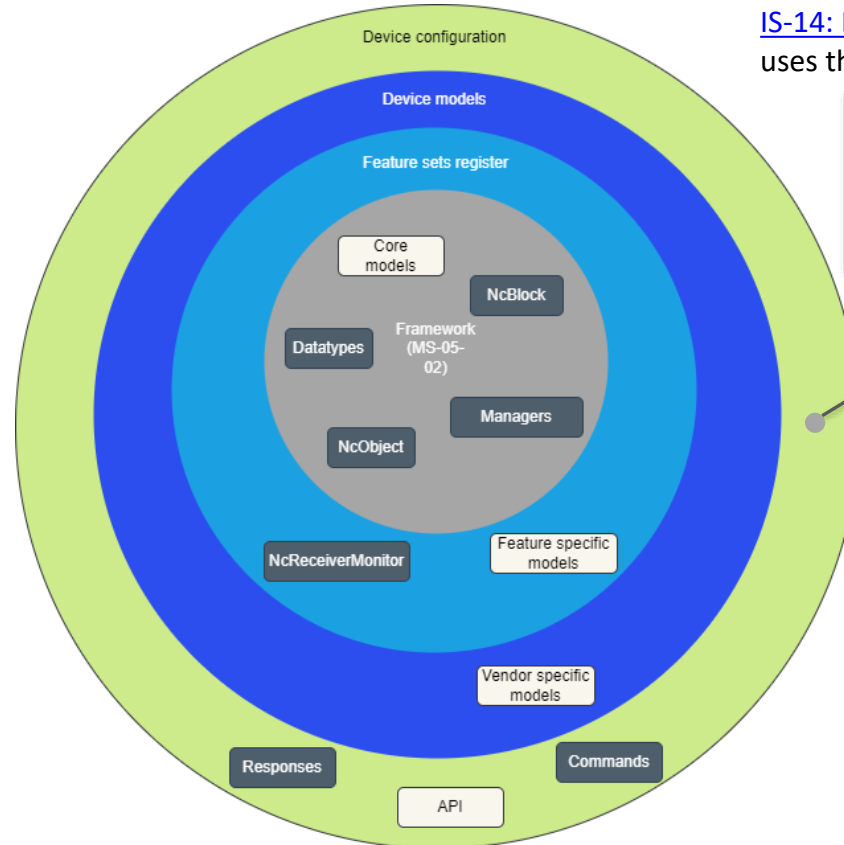


Modular

Creating a multi layered solution means we can mix and match to best address the target user stories.



Modular

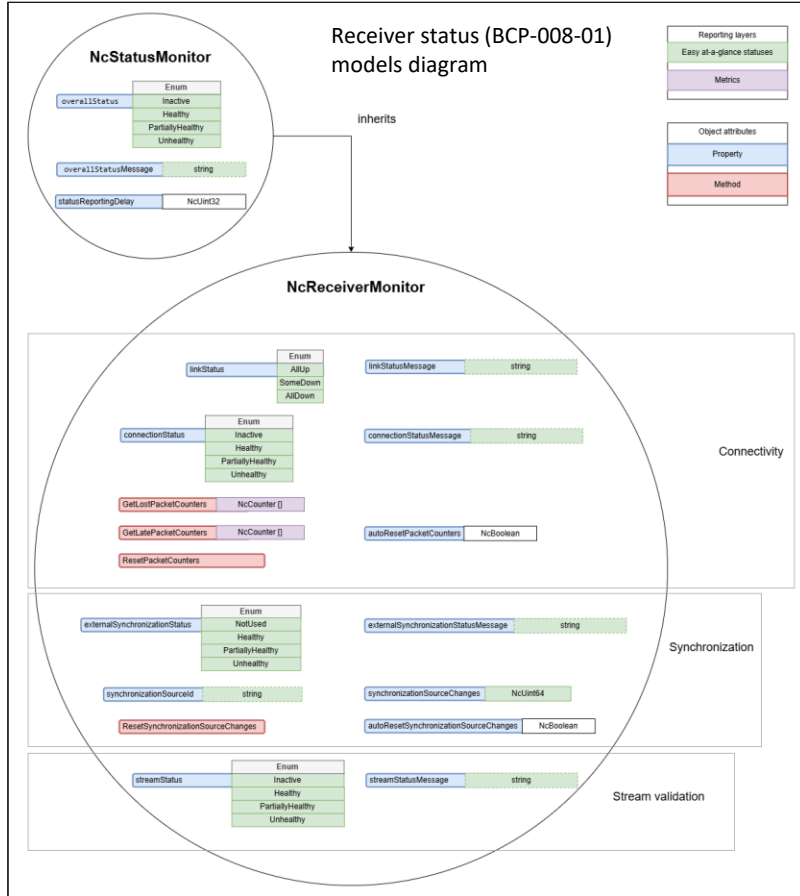


IS-14: Device configuration

uses the same underlying modelling language

HTTP based API with an emphasis on **retrieval** and **restoring** of configuration for backup, restore and other provisioning scenarios

Models solve problems



Industry bodies, system integrators, problem solvers need to be able to describe a problem and solution requirements using a modelling language which **feels natural**.

The solution models are published by a communication protocol which is an enabler.

Conformance strategy

